Solving a Multistage Transportation Problem Using LINGO

Anand Jayakumar A¹, Raghunayagan P²

¹ Department of Mechanical Engineering, SVS College of Engineering, Coimbatore, Tamil Nadu, India
² Department of Mechanical Engineering, Nehru Institute of Engineering and Technology, Coimbatore, Tamil Nadu, India

Abstract
Transhipment is the shipment of goods or containers to an intermediate destination, and then from there to yet another destination. One possible reason is to change the means of transport during the journey (for example from ship transport to road transport), known as transloading. Another reason is to combine small shipments into a large shipment (consolidation), dividing the large shipment at the other end (deconsolidation). Transshipment usually takes place in transport hubs. Much international transshipment also takes place in designated customs areas, thus avoiding the need for customs checks or duties, otherwise a major hindrance for efficient transport. Transshipment problems form a subgroup of transportation problems, where transshipment is allowed. In transshipment, transportation may or must go through intermediate nodes, possibly changing modes of transport. In this paper we consider a simple multistage transportation problem and solve it using LINGO package.

Keywords: Transhipment, shipment, goods, transloading, transport, LINGO.

1. Introduction
Transshipment or Transhipment is the shipment of goods or containers to an intermediate destination, and then from there to yet another destination. One possible reason is to change the means of transport during the journey (for example from ship transport to road transport), known as transloading. Another reason is to combine small shipments into a large shipment (consolidation), dividing the large shipment at the other end (deconsolidation). Transshipment usually takes place in transport hubs. Much international transshipment also takes place in designated customs areas, thus avoiding the need for customs checks or duties, otherwise a major hindrance for efficient transport. Transshipment problems form a subgroup of transportation problems, where transshipment is allowed. In transshipment, transportation may or must go through intermediate nodes, possibly changing modes of transport. The Transshipment problem has its origins in medieval times when trading started to become a mass phenomenon. Obtaining the minimum-cost route had been the main priority. However, technological development slowly gave priority to minimum-duration transportation problems. The classical transportation problem can be applied in a more general way in practice. Related problems as Multi-commodity transportation problem, Transportation problems with different kind of vehicles, Multi-stage transportation problems, Transportation problem with capacity limit is an extension of the classical transportation problem considering the additional special condition. For solving such problems many optimization techniques (dynamic programming, linear programming, special algorithms for transportation problem etc.) and heuristics approaches (e.g. evolutionary techniques) were developed. In this paper we consider a simple multistage transportation problem and solve it using LINGO package.

2. Literature Review

3. Mathematical Model

We consider the transportation of a single commodity through a multistage process. The items can be transported from factories to intermediate warehouses and then to retailers. In a two-stage transportation problem, we transport all the materials to warehouses from which they are transported to retailers.

Consider a two-stage transportation problem as follows:

\[ X_{ij} = \text{Quantity supplied from factory } i \text{ to warehouse } j \]
\[ Y_{jk} = \text{Quantity transported from warehouse } j \text{ to retailer } k \]
\[ C_{ij} = \text{Unit cost of transportation from factory } i \text{ to warehouse } j \]
\[ C_{jk} = \text{Unit cost of transportation from warehouse } j \text{ to retailer } k \]
\[ a_i = \text{Quantity available in factory } i \]
\[ b_k = \text{Quantity required by retailer } k \]

The objective is to minimize:

\[
\text{Minimize } \sum_i \sum_j C_{ij} X_{ij} + \sum_j \sum_k C_{jk} Y_{jk}
\]

Subject to:

\[
\sum_j X_{ij} \leq a_i \text{ where } i = 1, \ldots, m
\]
\[
\sum_i X_{ij} \geq \sum_k Y_{jk} \text{ where } j = 1, \ldots, p
\]
\[
\sum_j X_{ij} \geq b_k \text{ where } k = 1, \ldots, n
\]
\[ X_{ij}, Y_{jk} \geq 0 \]

4. The Problem

A two stage transportation problem with three factories (F1 to F3), two warehouses (W1 and W2) and three retailers (R1 to R3). The supplies in the factories are 100, 80 and 60 and the demands at the retailers are 90, 70 and 80. The warehouses have infinite capacity. The transportation cost is given in Table 1.

<table>
<thead>
<tr>
<th>No.</th>
<th>From-to</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>F1 – W1</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>F1 – W2</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>F2 – W1</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>F2 – W2</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>F3 – W1</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>F3 – W2</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>W1 – R1</td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td>W1 – R2</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>W1 – R3</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>W2 – R1</td>
<td>8</td>
</tr>
<tr>
<td>11</td>
<td>W2 – R2</td>
<td>6</td>
</tr>
<tr>
<td>12</td>
<td>W2 – R3</td>
<td>3</td>
</tr>
</tbody>
</table>

5. About LINGO

LINGO is a simple tool for utilizing the power of linear and nonlinear optimization to formulate large problems concisely, solve them, and analyze the solution. Optimization helps you find the answer that yields the best result; attains the highest profit, output, or happiness; or achieves the lowest cost, waste, or discomfort. Often these problems involve making the most efficient use of your resources—including money, time, machinery, staff, inventory, and more. Optimization problems are often classified as linear or nonlinear, depending on whether the relationships in the problem are linear with respect to the variables. LINGO includes a set of built-in solvers to tackle a wide variety of problems. Unlike many modeling packages, all of the LINGO solvers are directly linked to the modeling environment. This seamless integration allows LINGO to pass the problem to the appropriate solver directly in memory rather than through more sluggish intermediate files. This direct link also minimizes compatibility problems between the modeling language component and the solver components. Local search solvers are generally designed to search only until they have identified a local optimum. If the model is non-convex, other local optima may exist that yield significantly better solutions. Rather than stopping after the first local optimum is found, the Global solver will search until the global optimum is confirmed. The Global solver converts the original non-convex, nonlinear problem into several convex, linear subproblems. Then, it uses the branch-and-bound technique to exhaustively search over these subproblems for the global solution. The Nonlinear and Global license options are required to utilize the global optimization capabilities.
6. LINGO Program

Model:
\[ \text{Min} = 8X11 + 7X12 + 9X21 + 5X22 + 7X31 + 4X32 + 10Y11 + 5Y12 + 9Y13 + 8Y21 + 6Y22 + 3Y23; \]
\[ X11 + X12 \leq 100; \]
\[ X21 + X22 \leq 80; \]
\[ X31 + X32 \leq 60; \]
\[ Y11 + Y21 \geq 90; \]
\[ Y12 + Y22 \geq 70; \]
\[ Y13 + Y23 \geq 80; \]
\[ Y11 + Y12 + Y13 - X11 - X21 - X31 \leq 0; \]
\[ Y21 + Y22 + Y23 - X12 - X22 - X32 \leq 0; \]
\[ @GIN(X11);@GIN(X12);@GIN(X21);@GIN(X22); \]
\[ @GIN(X31);@GIN(X32); \]
\[ @GIN(Y11);@GIN(Y12);@GIN(Y13);@GIN(Y21); \]
\[ @GIN(Y22);@GIN(Y23); \]
End

7. Result and Discussion

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factory 1</td>
<td>Warehouse 1</td>
<td>70</td>
</tr>
<tr>
<td>Factory 1</td>
<td>Warehouse 2</td>
<td>30</td>
</tr>
<tr>
<td>Factory 2</td>
<td>Warehouse 2</td>
<td>80</td>
</tr>
<tr>
<td>Factory 3</td>
<td>Warehouse 3</td>
<td>60</td>
</tr>
<tr>
<td>Warehouse 1</td>
<td>Retailer 2</td>
<td>70</td>
</tr>
<tr>
<td>Warehouse 2</td>
<td>Retailer 1</td>
<td>90</td>
</tr>
<tr>
<td>Warehouse 2</td>
<td>Retailer 3</td>
<td>80</td>
</tr>
</tbody>
</table>

8. Conclusion

Thus we have solved the multistage transportation problem using linear programming method.

References